

What is claimed is:

1 1. An apparatus comprising:
2 a first plurality of queues, including a first queue and a second
3 queue, each of the first plurality of queues for holding a plurality of
4 pending memory requests;
5 one or more instruction-processing circuits, operatively coupled
6 to the plurality of queues, that inserts one or memory requests into at
7 least one of the queues based on a first instruction that specifies a
8 memory operation, and that inserts a first synchronization marker into
9 the first queue and inserts a second synchronization marker into the
10 second queue based on a second instruction that specifies a
11 synchronization operation, and that inserts one or memory requests into
12 at least one of the queues based on a third instruction that specifies a
13 memory operation; and
14 a first synchronization circuit, operatively coupled to the first
15 plurality of queues, that selectively halts processing of further memory
16 requests from the first queue based on the first synchronization marker
17 reaching a predetermined point in the first queue until the corresponding
18 second synchronization marker reaches a predetermined point in the
19 second queue.

1 2. The apparatus of claim 1, wherein the plurality of queues is
2 within a processor, and wherein the first queue is used for only vector

3 memory requests and synchronizations, and the second queue is used for
4 only scalar memory requests and synchronizations.

1 3. The apparatus of claim 2, wherein the first synchronization
2 instruction is an Lsync-type instruction.

1 4. The apparatus of claim 2, wherein the first synchronization
2 instruction is an Lsync V,S-type instruction.

1 5. The apparatus of claim 2, wherein, for a second synchronization
2 instruction, a corresponding synchronization marker is inserted to only
3 the first queue.

1 6. The apparatus of claim 5, wherein the second synchronization
2 instruction is an Lsync-type instruction.

1 7. The apparatus of claim 1, further comprising:
2 a second plurality of queues, including a third queue and a fourth
3 queue, each of the second plurality of queues for holding a plurality of
4 pending memory requests; wherein the plurality of queues is in a circuit
5 coupled to each of a plurality of processors including a first processor
6 and a second processor, and wherein the third queue is used for only
7 memory requests and synchronizations from the first processor, and the
8 second queue is used for only memory requests and synchronizations
9 from the second processor.

1 8. The apparatus of claim 7, further comprising:
2 a second synchronization circuit, operatively coupled to the
3 second plurality of queues, that selectively halts further processing of
4 memory requests from the third queue based on the first synchronization
5 marker reaching a predetermined point in the third queue until a
6 corresponding synchronization marker from the second processor
7 reaches a predetermined point in the fourth queue.

1 9. The apparatus of claim 1, further comprising:
2 a fifth queue for holding a plurality of write data elements,
3 wherein each write data element corresponds to a memory request in the
4 first queue, and wherein the write data elements are loaded into the fifth
5 queue decoupled from the loading of memory requests into the first
6 queue.

1 10. The apparatus of claim 1, further comprising:
2 a data cache; and
3 an external cache, wherein first synchronization instruction is an
4 Lsync V,S type instruction, preventing subsequent scalar references
5 from accessing the data cache until all vector references have been sent
6 to the external cache and all vector writes have caused any necessary
7 invalidations of the data cache.

1 11. A method comprising:

2 providing a first plurality of queues, including a first queue and a
3 second queue, each of the first plurality of queues for holding a plurality
4 of pending memory requests;
5 inserting one or memory requests into at least one of the queues
6 based on a first instruction that specifies a memory operation;
7 based on a second instruction that specifies a synchronization
8 operation inserting a first synchronization marker into the first queue
9 and inserting a second synchronization marker into the second queue;
10 inserting one or memory requests into at least one of the queues
11 based on a third instruction that specifies a memory operation;
12 processing memory requests from the first queue; and
13 selectively halting further processing of memory requests from
14 the first queue based on the first synchronization marker reaching a
15 predetermined point in the first queue until the corresponding second
16 synchronization marker reaches a predetermined point in the second
17 queue.

1 12. The method of claim 11, wherein the plurality of queues is within
2 a first processor, and wherein the inserting to first queue is for only
3 vector memory requests and synchronizations, and the inserting to the
4 second queue is for only scalar memory requests and synchronizations.

1 13. The method of claim 12, wherein the first synchronization
2 instruction is an Lsync-type instruction.

1 14. The method of claim 12, wherein the first synchronization
2 instruction is an Lsync V,S-type instruction.

1 15. The method of claim 12, wherein based on a second
2 synchronization instruction that specifies a synchronization operation,
3 inserting a corresponding synchronization marker to only the first
4 queue.

1 16. The method of claim 15, wherein the second synchronization
2 instruction is an Lsync-type instruction.

1 17. The method of claim 11, further comprising:
2 providing a plurality of processors including a first processor and
3 a second processor;
4 providing a second plurality of queues, including a third queue
5 and a fourth queue, each of the second plurality of queues for holding a
6 plurality of pending memory requests;
7 inserting to the third queue only memory requests and
8 synchronizations from the first processor; and
9 inserting to the second queue only memory requests and
10 synchronizations from the second processor.

1 18. The method of claim 17, further comprising:
2 selectively halting further processing of memory requests from
3 the third queue based on the first synchronization marker reaching a
4 predetermined point in the third queue until a corresponding

1 synchronization marker from the second processor reaches a
2 predetermined point in the fourth queue.

1 19. The method of claim 11, further comprising:
2 queueing a plurality of write data elements to a fifth queue,
3 wherein each write data element corresponds to a memory request in the
4 first queue, and wherein the write data elements are inserted into the
5 fifth queue decoupled from the inserting of memory requests into the
6 first queue.

1 20. The method of claim 11, further comprising:
2 providing a data cache and an external cache, wherein first
3 synchronization instruction is an Lsync V,S type instruction; and
4 preventing subsequent scalar references from accessing the data
5 cache until all vector references have been sent to the external cache and
6 all vector writes have caused any necessary invalidations of the data
7 cache based on the first synchronization instruction.

1 21. An apparatus comprising:
2 a first plurality of queues, including a first queue and a second
3 queue, each of the first plurality of queues for holding a plurality of
4 pending memory requests;
5 means for inserting one or memory requests into at least one of
6 the queues based on a first instruction that specifies a memory
7 operation;

8 means for, based on a second instruction that specifies a
9 synchronization operation inserting a first synchronization marker into
10 the first queue and inserting a second synchronization marker into the
11 second queue;
12 means for inserting one or memory requests into at least one of
13 the queues based on a third instruction that specifies a memory
14 operation;
15 means for processing memory requests from the first queue; and
16 means for selectively halting further processing of memory
17 requests from the first queue based on the first synchronization marker
18 reaching a predetermined point in the first queue until the corresponding
19 second synchronization marker reaches a predetermined point in the
20 second queue.

1 22. The apparatus of claim 21, wherein the plurality of queues is
2 within a first processor, and wherein the means for inserting to first
3 queue operates for only vector memory requests and synchronizations,
4 and the means for inserting to the second queue operates for only scalar
5 memory requests and synchronizations.

1 23. The apparatus of claim 22, wherein the first synchronization
2 instruction is an Lsync-type instruction.